WE CLAIM:

1   1.   A method, comprising:

2   receiving a driver package;

3   generating a strong name for the driver package; and

4   assigning a storage location for driver files associated with the driver

5 package based on the strong name.

1   2.   A method according to Claim 1,

2   wherein the driver files include a driver image file, and

3   wherein the method further comprises loading the driver image file from

4 the storage location into a memory.

1   3.   A method according to Claim 1, wherein the driver files include at

2 least one of a driver image file and a service name.

1   4.   A method according to Claim 1, wherein generating the strong name

2 for the driver package includes hashing at least one of the driver files associated with the

3 driver package.

1   5.   A method according to Claim 4, wherein the at least one driver file

2 includes any one of a driver catalog file, a setup information file, and a driver image file.

1   6.   A method according to Claim 4, wherein generating the strong name

2   for the driver package includes concatenating the hash of the at least one driver file to at

3   least one of a driver catalog file and a public key from the driver package.

1   7.   A method according to Claim 1, wherein assigning a storage location

2   to driver files from the driver package in accordance with the strong name includes

3   assigning at least one driver file associated with the driver package to a subdirectory in a

4   driver store.

1   8.   A method according to Claim 1, wherein assigning a storage location

2   to data from the driver package in accordance with the strong name includes assigning

3   service name parameters to a service name database.

1   9.   A method, comprising:

2   generating a unique identity for respective driver packages to which

3   commonly named driver files correspond; and

4   assigning driver files associated with the individual driver packages to

5   subdirectories in a common storage based on the unique identity.

1   10.   A method according to Claim 9,

2   wherein the driver files include a driver, and

3   wherein the method further comprises loading the driver from one of the

4   subdirectories into a memory.

1   11.   A method according to Claim 9, wherein the driver files include at

2   least one of a driver image file and a service name.

1   12.   A method according to Claim 9, wherein generating the unique

2   identity for the respective driver packages to which commonly named driver files

3   correspond includes hashing at least one driver file associated with the driver package.

1   13.   A method according to Claim 12, wherein generating the unique

2   identity for the respective driver packages to which commonly named driver files

3   correspond includes appending a vendor's digital signature to the hash.

1   14.   A method according to Claim 9, wherein assigning driver files from

2   the respective driver packages to subdirectories in a common storage avoids overwriting

3   previous versions of commonly named driver files.

1   15.   A method according to Claim 11, wherein the assigning includes

2   assigning driver image files from the respective driver packages to subdirectories in a

3   driver store based on the unique identity for the driver package to which the respective

4   driver image files correspond.

1   16.   A method according to Claim 11, wherein the assigning includes

2   assigning service names from the respective driver packages to a service name database

3   that includes a service key and an image path corresponding to one of the driver files.

1   17.   A computer-readable medium having one or more instructions to be

2 executed by one or more processors, the one or more instructions causing the one or more

3 processors to:

4   generate a strong name for a driver package;

5   assign a storage location for a driver file associated with the driver package

6 based on the strong name.

1   18.   A computer-readable medium according to Claim 17,

2   wherein the driver file is a driver image file, and

3   wherein the one or more instructions cause the one or more processors to

4 further load the driver from the storage location into a memory.

1   19.   A computer-readable medium according to Claim 17, wherein the

2 strong name is a hash of at least one driver file associated with the driver package.

1   20.   A computer-readable medium according to Claim 17, wherein the

2 strong name incorporates at least one of a driver catalog file and a public key

3 corresponding to a vendor of the driver package.

1   21.   A computer-readable medium according to Claim 17, wherein the

2 driver file is at least one of a driver image file and a co-installer, and wherein further the

3 storage location is a driver store subdirectory.

1   22.   A computer-readable medium according to Claim 17, wherein the

2   driver file is a service name, and wherein further the storage location is database to store

3   a corresponding service key and image path.

1   23.   A computer-readable medium according to Claim 17, wherein the

2   one or more instructions causing the one or more processors to assign a storage location

3   for the driver file associated with the driver package refrains from assigning previously

4   assigned storage locations.

1   24.   An apparatus, comprising:

2   an installer to generate a strong name for a driver package; and

3   a storage to install files associated with the driver package based on the

4   strong name.

1   25.   An apparatus according to Claim 24,

2   wherein the files associated with the driver package include a driver, and

3   wherein the apparatus further includes a loader to load the driver into a

4   memory.

1   26.   An apparatus according to Claim 24, wherein the installer is to

2   generate the strong name as a hash function of at least one driver file related to the driver

3   package.

1   27.   An apparatus according to Claim 24, wherein the installer is to

2 generate the strong name by incorporating at least one of a digital signature and a public

3 key related to the driver package.

1   28.   An apparatus according to Claim 24, wherein the storage includes a

2 subdirectory corresponding to the strong name.

1   29.   An apparatus according to Claim 28, wherein the storage is a driver

2 store, and wherein further the files include at least one of a driver image file and a co-

3 installer.

1   30.   An apparatus according to Claim 28, wherein the storage is a

2 database, and wherein further the files include a service name having a corresponding

3 service key and image path.

1   31.   An apparatus, comprising:

2   means for generating a strong name for a driver package; and

3   means for storing files associated with the driver package based on the

4 strong name.

1   32.   An apparatus according to Claim 31,

2   wherein the files associated with the driver package include a driver image

3 file, and

4   wherein the apparatus further includes means for loading the driver image

5   file into a memory.

1   33.   An apparatus according to Claim 31, wherein the means for

2   generating hashes at least one driver file from the driver package.

1   34.   An apparatus according to Claim 31, wherein means for generating

2   incorporates a digital signature related to the driver package into the strong name.

1   35.   An apparatus according to Claim 31, wherein the means for storing

2   has a subdirectory corresponding to the strong name.

1   36.   An apparatus according to Claim 34, wherein the means for storing

2   has a subdirectory for storing at least one of a driver package, a driver image file, and a

3   co-installer.

1   37.   An apparatus according to Claim 34, wherein the means for storing

2   stores a service name having a corresponding service key and image path.

1   38.   An apparatus according to Claim 34, wherein the means for storing

2   stores a pointer from an operating system to a driver file for a particular device.